

Amendments to the Claims:

This listing of claims will replace all prior version, and listings, of claims in the application:

Listing of Claims:

1-11. (canceled)

12. (currently amended) A method comprising:

instantiating a distributed software environment that includes two or more physical processing elements and a runtime debugging architecture;

executing first and second software programs in respective first and second physical processing elements within the distributed software environment;

generating event records during execution of the first and second software programs, in response to occurrences of events;

receiving at least some of the event records at the runtime debugging architecture of the distributed software environment;

forwarding at least some of the event records from the distributed software environment to a debugging host outside of the distributed software environment;
and

providing a simulation of the distributed software environment at the debugging host, wherein the simulation includes state changes based at least in part on one or more of the event records received from the distributed software environment; and

wherein the operation of providing a simulation of the distributed software environment comprises simulating the distributed software environment at the debugging host substantially simultaneously with execution of the first and second software programs in the distributed software environment.

13. (canceled)

14. (previously presented) A method according to claim 12, further comprising:
collecting event records at a runtime system within the distributed software environment;

forwarding event records from the runtime system to the runtime debugging architecture; and

forwarding event records from the runtime debugging architecture to the debugging host along a communication channel.

15. (previously presented) A method according to claim 12, wherein:

the runtime debugging architecture adds time stamps to event records; and

the runtime debugging architecture adds causality stamps to event records, to identify causes of events associated with the event records.

16. (previously presented) A method according to claim 12, wherein the operation of forwarding event records to a debugging host comprises forwarding event records to the debugging host via an intermediate processing element.

17. (previously presented) A method according to claim 16, wherein the operation of forwarding event records to the debugging host via an intermediate processing element comprises storing one or more of the event records in a flash memory before forwarding the event records to the debugging host.

18. (previously presented) A method according to claim 12, wherein the operation of generating event records comprises generating an event record in response to at least one occurrence selected from the group consisting of:

an occurrence of an event selected for logging according to a predetermined design model associated with the distributed software environment; and

execution of an explicit event recording call included in one of the software programs.

19. (previously presented) A method according to claim 12, further comprising:

monitoring, with a probe, a bus trace associated with at least one of the physical processing elements; and

wherein the operation of generating event records comprises generating event records in response to activity detected on the bus trace.

20. (previously presented) A method according to claim 12, further comprising:

transmitting an event token from the distributed software environment to the debugging host;

expanding the event token at the debugging host, based at least in part on a predetermined sequence of events associated with the event token; and

simulating the distributed software environment, based at least in part on the predetermined sequence of events associated with the event token.

21. (previously presented) A method according to claim 12, wherein the operation of providing a simulation of the distributed software environment comprises:

tracking state changes that occur within a simulated environment in the debugging host, the state changes based at least in part on event records received from the distributed software environment.

22. (currently amended) An apparatus comprising:

a machine accessible medium; and
instructions encoded in the machine accessible medium which, when
executed, perform operation comprising:
receiving, at a debugging host, event records relating to events generated
during execution of multiple software programs in different physical processing
elements of a distributed software environment; and
providing a simulation of the distributed software environment at the
debugging host, based at least in part on one or more of the received event records;
and

wherein the operation of providing a simulation of the distributed software
environment comprises simulating the distributed software environment at the
debugging host substantially simultaneously with execution of the first and second
software programs in the distributed software environment.

23. (previously presented) An apparatus according to claim 22, wherein the
operation of receiving event records at the debugging host comprises:

receiving event records that include time stamps and causality stamps,
wherein the causality stamps identify causes of the events associated with the event
records.

24. (currently amended) A system comprising:

- multiple physical processing elements;
- a distributed software environment that supports the multiple physical processing elements;
- software programs within the distributed software environment and associated with the physical processing elements, wherein the software programs, when executed, trigger events;
- a runtime system that collects event records pertaining to events triggered by the software programs; and
- a runtime debugging architecture that receives the event records from the runtime system and forwards at least some of the event records to a debugging host, to allow the debugging host to simulate the distributed software environment substantially simultaneously with execution of the software programs in the distributed software environment, based at least in part on one or more of the event records.

25. (previously presented) A system according to claim 24, wherein:

- the runtime debugging architecture adds time stamps to at least some of the event records received from the runtime system; and
- the runtime debugging architecture adds causality stamps to at least some of the event records received from the runtime system, wherein the causality stamps identify causes of the events associated with the event records.

26. (previously presented) A system according to claim 24, wherein the distributed software environment generates an event record in response to at least one occurrence selected from the group consisting of:

- an occurrence of an event selected for logging according to a predetermined design model associated with the distributed software environment; and
- execution of an explicit event recording call included in at least one of the software programs.

27. (currently amended) ~~An apparatus~~ A system according to claim 24, further comprising:

a probe to monitor a bus trace associated with at least one of the physical processing elements; and

wherein the distributed software environment generates an event record in response to activity detected on the bus trace by the probe.

28. (currently amended) A debugging host comprising:

a communication interface to receive event records from a distributed software environment external to the debugging host, wherein the event records relate to events generated during execution of software programs within the distributed software environment; and

a simulator to simulate the distributed software environment substantially simultaneously with execution of the software programs in the distributed software environment, based at least in part on one or more of the event records received from the distributed software environment;

wherein the simulator tracks state changes within the simulated distributed software environment, based at least in part on one or more of the event records received from the distributed software environment.

29. (previously presented) A debugging host according to claim 28, wherein:

in response to the distributed software environment forwarding an event token to the debugging host, the simulator performs operation comprising:

expanding the event token, based at least in part on a predetermined sequence of events associated with the event token; and

simulating the distributed software environment, based at least in part on the predetermined sequence of events associated with the event token.